

Análise e Visualização de Dados 2023/2024

Diário de Bordo de 14 de março de 2024 (quinta-feira)

Realizado por Renata Martins - PG52896

Sumário

- Diretrizes para avaliação dos alunos
- SpaCy
 - Apresentação
 - Downloads e Testes (passo-a-passo)

- Diretrizes para avaliação dos alunos

Durante a primeira parte da aula, o professor Álvaro apresentou o ficheiro “Contributos para o enunciado do trabalho final de AVD + PI2-sombra” disponível na Blackboard.

O documento descreve as diretrizes para o trabalho final das disciplinas de Análise e Visualização de Dados e Projeto Integrado 2, que incide sobre a análise de entrevistas. Os alunos devem realizar várias tarefas de processamento de texto e análise de dados, como extração de entidades nomeadas, lematização, análise de sentimentos, entre outras, usando ferramentas como expressões regulares, SpaCy, Python e Excel. Os trabalhos finais incluem a geração de um PDF e a proposta de uma "História (Alternativa) da Universidade" baseada nos dados analisados. São dadas orientações específicas sobre a estrutura do relatório, a redação, a citação de fontes e a apresentação dos resultados. Atenção! É necessário fazer uma retificação ao documento: o trabalho será em grupo e o relatório também. Portanto, onde se lê “*Apresentação de relatório individual*”, considerar “*Apresentação de relatório do grupo*”.

- SpaCy

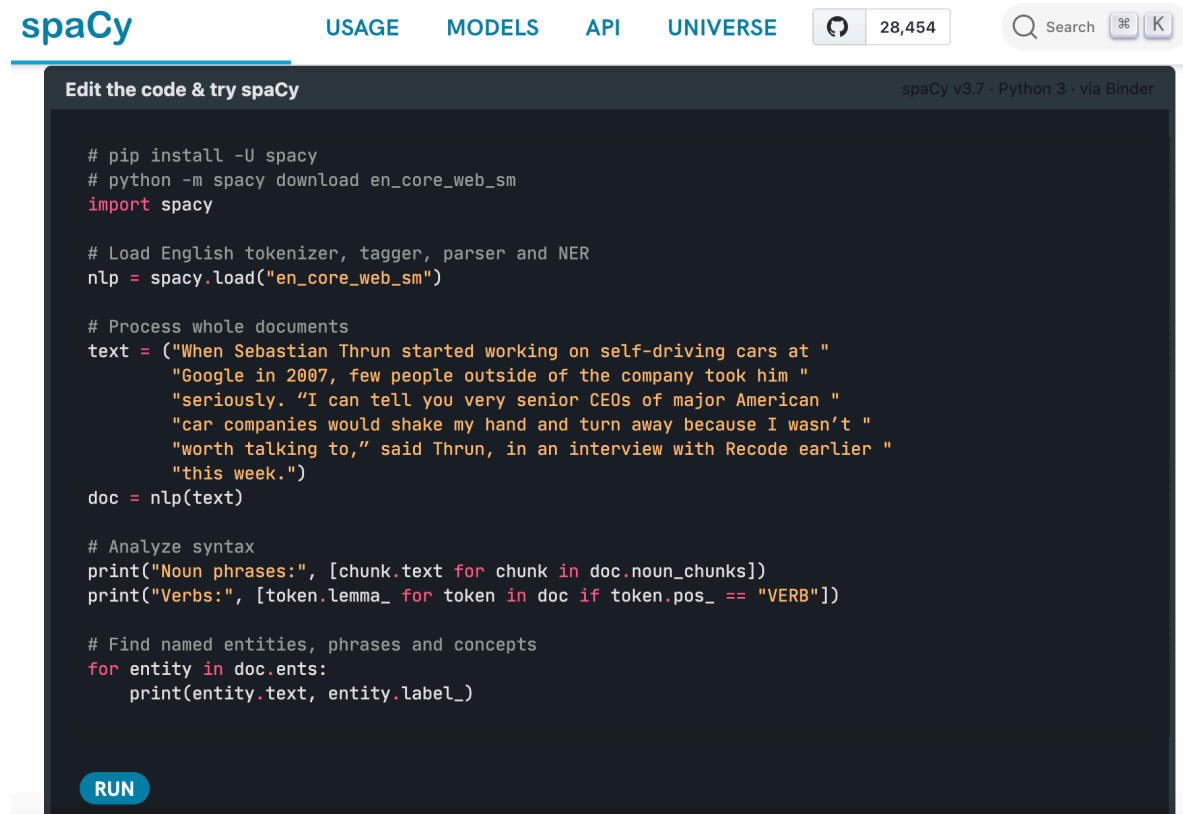
O professor JJ apresentou a SpaCy - uma biblioteca de software livre e de código aberto, projetada para trabalhar com processamento de linguagem natural.

O que essa biblioteca faz? Ajuda a realizar tarefas com texto, como identificar partes do discurso em uma frase (por exemplo, distinguir substantivos de verbos), reconhecer nomes de pessoas ou lugares.

Prática:

1 - Aceder ao Spacy

Vá até o sítio da SpaCy no link: <https://spacy.io>, copie o código disponível na página principal e cole em novo documento .py para editar os códigos.



2 - Instalar o spacy no terminal

```
pip install -U spacy
```

No Mac costuma dar jeito com: `python3 -m pip install spacy`

3 - Importar a biblioteca SpaCy para o seu script ou ambiente de trabalho em Python

```
import spacy
```

4 - Fazer download do modelo de linguagem para o português

Denominado **pt_core_news_lg**. Este modelo é necessário para que a biblioteca entenda a linguagem e possa realizar tarefas, como tokenização, análise

morfológica, reconhecimento de entidades nomeadas, etc.

Para isso, no substitua `en_core_web_sm` (modelo para inglês) por `pt_core_news_lg`

```
# Load Portuguese tokenizer, tagger, parser and NER
nlp = spacy.load("pt_core_news_lg")
```

5 - Processar um texto em português

```
# Process whole documents
text = """Bom dia, Alvaro Iriarte da Silva. Onde é que nasceu?
Em Viana do Castelo."""
doc = nlp(text)
```

Aqui é onde a mágica acontece. Ao chamar `nlp(text)`, você está passando a string `text` para o modelo de NLP para processá-la.

6 - Fazer a análise sintática

```
# Analyze syntax
# print("Noun phrases:", [chunk.text for chunk in
doc.noun_chunks])
Esta linha imprime as frases nominais (grupos de palavras que funcionam como substantivos). doc.noun_chunks gera uma sequência de tais frases no documento.
# print("Verbs:", [token.lemma_ for token in doc if token.pos_
== "VERB"])
```

Aqui, o código vai extrair e imprimir os verbos do texto. Ele verifica se a classe gramatical (`pos_`) é "VERB". Se for, ele pega o lema do verbo (`lemma_`), que é a forma base da palavra (por exemplo, "nascer" para "nasceu").

7 - Encontrar entidades nomeadas, frases e conceitos

```
# Find named entities, phrases and concepts
for ent in doc.ents:
    print(ent.text, ent.label_)
```

O comando vai percorrer todas as entidades nomeadas (nomes de pessoas, lugares, organizações, datas) detectadas no texto.

Ao dar print, para cada entidade, o código imprime o texto da entidade (`ent.text`) e seu rótulo (`ent.label_`), que indica o tipo da entidade (por exemplo, "PESSOA", "LOCAL").

8 - Extrair informações detalhadas sobre cada palavra

```
for fr in doc.sents:
    print(f"====={fr.text}")
    for pal in fr:
        # print(f"{pal.text}, pos={pal.pos}, lemma={pal.morph}
        == {pal.rank}")
        print(f"{pal.text}, pos={pal.pos},
        lemma={pal.lemma_}")
    print(f"{pal.text}\t{pal.pos_}\t{pal.lemma_}\t{pal.rank}")
```

Para cada palavra, o comando imprime o texto da palavra, sua classe gramatical (pos_), a forma lematizada (lemma_), e o ranking (rank), que pode ser usado para determinar a frequência da palavra na biblioteca.

```
====Bom dia, Álvaro Iriarte da Silva.
Bom, pos=ADJ, lemma=bom, morf=Gender=Masc|Number=Sing == 1333
dia, pos=NOUN, lemma=dia, morf=Gender=Masc|Number=Sing == 67
., pos=PUNCT, lemma=., morf= == 0
Álvaro, pos=PROPN, lemma=Álvaro, morf=Gender=Masc|Number=Sing == 9000
Iriarte, pos=PROPN, lemma=Iriarte, morf=Number=Sing == 307392
da, pos=ADP, lemma=de o, morf=Definite=Def|Gender=Fem|Number=Sing|PronType=Art == 9
Silva, pos=PROPN, lemma=Silva, morf=Number=Sing == 747
., pos=PUNCT, lemma=., morf= == 3
====Onde onde é que nasceu?
Onde, pos=PRON, lemma=onde, morf=PronType=Rel == 3308
onde, pos=PRON, lemma=onde, morf=Gender=Fem|Number=Sing|PronType=Rel == 92
é, pos=AUX, lemma=ser, morf= == 14
que, pos=SCONJ, lemma=que, morf= == 7
nasceu, pos=VERB, lemma=nascer, morf=Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm
=Fin == 1612
?, pos=PUNCT, lemma=?, morf= == 49
====Em Viana do Castelo.
Em, pos=ADP, lemma=em, morf= == 76
Viana, pos=PROPN, lemma=Viana, morf=Gender=Fem|Number=Sing == 6390
do, pos=ADP, lemma=de o, morf=Definite=Def|Gender=Masc|Number=Sing|PronType=Art == 8
Castelo, pos=PROPN, lemma=Castelo, morf=Number=Sing == 3950
., pos=PUNCT, lemma=., morf= == 3
zdt:HD$ █
```

Para organizar as saídas em colunas, o professor usou a tabulação \t para separar as informações, o que alinhou os textos verticalmente.

```
print(f"{pal.text}\t{pal.pos_}\t{pal.lemma_}\t{pal.rank}")
```

Resultado gerado:

```

-TRT Warning: Could not find TensorRT
zdt:HD$ cat out
Álvaro Iriarte da Silva PER
Viana do Castelo LOC
====Bom dia, Álvaro Iriarte da Silva.
Bom      ADJ      bom      1333
dia      NOUN     dia      67
        PUNCT     ,        0
Álvaro   PROPON   Álvaro   9000
Iriarte   PROPON   Iriarte  307392
da        ADP      de o     9
Silva     PROPON   Silva    747
        PUNCT     .        3
====Onde onde é que nasceu?
Onde      PRON     onde     3308
onde      PRON     onde     92
é         AUX      ser      14
que       SCONJ    que      7
nasceu    VERB     nascer   1612
?         PUNCT    ?        49
====Em Viana do Castelo.
Em        ADP      em       76
Viana     PROPON   Viana    6390
do        ADP      de o     8
Castelo   PROPON   Castelo  3950
        PUNCT     .        3
zdt:HD$ gnumeric out

(gnumeric:86514): IBUS-WARNING **: 17:48:30.084: Unable to connect to ibus: Could not
connect: Connection refused
zdt:HD$

```

Quadro final dos comandos da aula:

```

1 # pip install -U spacy
2 # python -m spacy download pt_core_news_lg (português)
3 # en_core_web_sm (inglês)
4 import spacy
5
6 # Load English tokenizer, tagger (associar categoria morfosintática), parser (calcular a árvore sintática) and NER (entidades).
7 nlp = spacy.load("pt_core_news_lg")
8
9 # Process whole documents
10 text = ("Bom dia, Álvaro Iriarte da Silva. Onde é que nasceu? Em Viana do Castelo.")
11 doc = nlp(text)
12
13 # Analyze syntax
14 # print("Noun phrases:", [chunk.text for chunk in doc.noun_chunks])
15 # print("Verbs:", [token.lemma_ for token in doc if token.pos_ == "VERB"])
16
17 # Find named entities, phrases and concepts
18 for ent in doc.ents:
19     print(ent.text, ent.label_)
20
21 for fr in doc.sents:
22     print(f"===={fr.text}")
23     for pal in fr:
24         # print(f"{pal.text}, pos={pal.pos}, lemma={pal.morph} == {pal.rank}")
25         print(f"{pal.text}\t{pal.pos_}\t{pal.lemma_}\t{pal.rank}")
26
27

```

